



JP2001067187

Biblio Page 1

esp@cenet

**STORAGE SUB-SYSTEM AND ITS CONTROL METHOD**

Patent Number: JP2001067187
Publication date: 2001-03-16
Inventor(s): ARAKAWA TAKASHI; MOGI KAZUHIKO; YAMAKAMI KENJI; ARAI HIROHARU
Applicant(s):: HITACHI LTD
Requested Patent: ☐ JP2001067187 (JP01067187)
Application Number: JP19990242713 19990830
Priority Number(s):
IPC Classification: G06F3/06 ; G06F12/00
EC Classification:
Equivalents:

Abstract

PROBLEM TO BE SOLVED: To simplify a work for optimizing arrangement by re-arrangement by the user of a disk array system or the like by changing the correspondence of a logical storage area from a physical storage area into the second physical storage area and executing re-arrangement.

SOLUTION: A control part 300 automatically executes re-arrangement execution processing at the set time and date. That is, the part 300 copies contents stored in a re-arrangement source physical area in a re-arrangement destination physical area based on re-arrangement information 408. Moreover, at the point of time when the copying is completed and the whole contents of the re-arrangement source physical area are reflected in the re-arrangement destination physical area, the control part 300 changes a physical area corresponding to a logical area for executing re-arrangement in logical/physical correspondence information 400 from the re-arrangement source physical area into the re-arrangement destination physical area. Besides, the control part 300 uses the re-arrangement destination physical area on a non-usage physical area 1470, changes the re-arrangement source physical area into the non-usage one and, moreover, updates the time and date of re-arrangement execution time information 406 into the one for a next time by referring to time and date updating information on re-arrangement execution time information 406.

Data supplied from the esp@cenet database - 12

(19) 日本特許庁 (J P) (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-67187
(P2001-67187A)

(43) 公開日 平成13年3月16日 (2001.3.16)

(51) Int.Cl.	G06F 3/06	3/06	FI	G06F 3/06	3/06	301A	5B065	540	5B082	501B
(52) 優先権	3/01	540	501	12/00	12/00	301A	5B065	540	5B082	501B

審査請求 未請求 請求項の数10 O L (全 24 頁)

(21) 出願番号	特願平11-242713	(71) 出願人	000005108
(22) 出願日	平成11年8月30日 (1999.8.30)	株式会社日立製作所	株式会社日立製作所
		東京千代田区神田墨田区西四丁目6番地	東京千代田区神田墨田区西四丁目6番地
		荒川 敬史	荒川 敬史
		株式会社日立製作所システム開発研究所内	株式会社日立製作所システム開発研究所内
		神奈川県川崎市麻生区王禅寺1099番地 株	神奈川県川崎市麻生区王禅寺1099番地 株
		茂木 和彦	茂木 和彦
		株式会社日立製作所システム開発研究所内	株式会社日立製作所システム開発研究所内
		100075096	100075096
		弁理士 作田 康夫	弁理士 作田 康夫

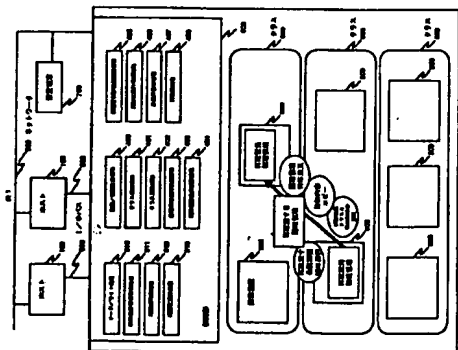
最終頁に続く

(54) 【発明の名称】 ストレージサブシステム及びその制御方法

(57) 【要約】

【課題】 ストレージサブシステムのユーザまたは保守員が記憶領域の物理的再配置による配置最適化を行うための作業を簡便にするストレージサブシステムおよび制御方法を提供する。

【解決手段】 ストレージサブシステム200は、記憶装置500を、それぞれ属性を有する複数の組 (クラス) 600として管理し、クラス属性に基づき好適な再配置先のクラスを決定する。



【特許請求の範囲】

【請求項1】 複数の記憶装置と、前記記憶装置の使用状況情報を取得する手段と、前記計算機がリードライト対象とする記憶装置と前記記憶装置の第一の物理記憶領域とを対応づける手段とを有し、1台以上の計算機に接続するストレージサブシステムの制御方法であって、前記記憶装置は複数の組 (クラス) に分類され、前記クラスは設定された属性を有し、前記ストレージサブシステムは、前記使用状況情報および前記クラス属性に基づき前記記憶装置の再配置先として利用可能な第二の物理記憶領域を前記クラスの中から選択し、前記第一の物理記憶領域の内容を前記第二の物理記憶領域にコピーするとともに前記記憶領域の対応づけを前記第一の物理記憶領域から前記第二の物理記憶領域へ変更して再配置を行うことを特徴とするストレージサブシステムの制御方法。

【請求項2】 請求項1に記載のストレージサブシステムの制御方法であって、ストレージサブシステムは、前記使用状況情報を蓄積し、設定された期間の前記使用状況情報に基づき、論理記憶領域の再配置先を決定し、設定された時間に再配置を行うことを特徴とするストレージサブシステムの制御方法。

【請求項3】 請求項1または2に記載のストレージサブシステムの制御方法であって、ストレージサブシステムは、使用状況情報として、記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスは、属性として設定されたクラス間の性能順位と使用率上限値を有し、前記ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

【請求項4】 請求項1または2に記載のストレージサブシステムの制御方法であって、ストレージサブシステムは、使用状況情報として、記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスは、属性として設定されたクラス間の性能順位と使用率上限値を有し、前記ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

【請求項5】 請求項1または2に記載のストレージサブシステムの制御方法であって、ストレージサブシステムは、使用状況情報として、記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスは、属性として設定されたクラス間の性能順位と使用率上限値を有し、前記ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

【請求項6】 請求項1または2に記載のストレージサブシステムの制御方法であって、ストレージサブシステムは、使用状況情報として、記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスは、属性として設定されたクラス間の性能順位と使用率上限値を有し、前記ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域に対するアクセス種別の分析結果に基づいて前記論理記憶領域の再配置先のクラスを前記対象アクセス種別のクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

【請求項6】 1台以上の計算機に接続し、複数の記憶装置と、前記記憶装置の使用状況情報を取得する手段と、前記計算機がリードライト対象とする記憶装置と前記記憶装置の第一の物理記憶領域との対応づけを行う手段とを有するストレージサブシステムであって、前記複数のディスク装置をそれぞれ属性を有する複数の組 (クラス) として管理する手段と、前記使用状況情報および前記クラス属性に基づき前記記憶領域に好適な再配置先のクラスを決定する手段と、前記論理記憶領域の再配置先として利用可能な第二の物理記憶領域を前記クラスの中から選択し、前記第一の物理記憶領域の内容を前記第二の物理記憶領域にコピーするとともに前記記憶領域の対応づけを前記第一の物理記憶領域から前記第二の物理記憶領域へ変更して再配置を行う手段とを有することを特徴とするストレージサブシステム。

【請求項7】 請求項6に記載のストレージサブシステムであって、ストレージサブシステムは、前記使用状況情報を蓄積し、設定された期間の前記使用状況情報に基づき、論理記憶領域の再配置先を自動的に決定する手段と、設定された時間に再配置を行う手段とを有することを特徴とするストレージサブシステム。

【請求項8】 請求項6または7に記載のストレージサブシステムであって、ストレージサブシステムは、使用状況情報として記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスに属性として設定されている使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

【請求項9】 請求項6または7に記載のストレージサブシステムであって、ストレージサブシステムは、使用状況情報として、記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスは、属性として設定されたクラス間の性能順位と使用率上限値を有し、前記ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

【請求項10】 請求項6または7に記載のストレージサブシステムの制御方法であって、ストレージサブシステムは、使用状況情報として、記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスは、属性として設定されたクラス間の性能順位と使用率上限値を有し、前記ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

【請求項11】 請求項6または7に記載のストレージサブシステムの制御方法であって、ストレージサブシステムは、使用状況情報として、記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスは、属性として設定されたクラス間の性能順位と使用率上限値を有し、前記ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

【請求項12】 請求項6または7に記載のストレージサブシステムの制御方法であって、ストレージサブシステムは、使用状況情報として、記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスは、属性として設定されたクラス間の性能順位と使用率上限値を有し、前記ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

【請求項13】 請求項6または7に記載のストレージサブシステムの制御方法であって、ストレージサブシステムは、使用状況情報として、記憶装置の単位時間当たりの使用時間 (使用率) を用い、各クラスは、属性として設定されたクラス間の性能順位と使用率上限値を有し、前記ストレージサブシステムは、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域を選択し、前記論理記憶領域の再配置先のクラスを前記順位の高いクラスから、各クラスの使用率上限値を超えないように決定することを特徴とするストレージサブシステムの制御方法。

るストレージサブシステム。

【請求項10】請求項6、7、8、または9に記載のストレージサブシステムであって、ストレージサブシステムは、複数のディスク装置を有するディスクアレイであり、前記ディスク装置の使用率を使用状況情報として用いる手段を有することを特徴とするストレージサブシステム。

【発明の詳細な説明】

【0001】
【発明の属する技術分野】本発明は、複数の記憶装置を有するストレージサブシステム、およびその制御方法に関する。

【0002】

【従来の技術】コンピュータシステムにおいて、高性能を実現する二次記憶システムの1つにディスクアレイシステムがある。ディスクアレイシステムは、複数のディスク装置をアレイ状に配置し、前記各ディスク装置に分割格納されるデータのリード/ライトを、前記各ディスク装置を並列に動作させることによって、高速に行うシステムである。ディスクアレイシステムに関する論文としては、D. A. Patterson, G. G. Gibson, and R. H. Kats, "A Case for Redundant Arrays of inexpensive Disks (RAID)" (in Proc. ACM SIGMOD, pp. 109-116, June 1988) がある。この論文では、冗長性を付加したディスクアレイシステムに対し、その構成に応じてレベル1からレベル5の種別を与えている。これらの種別に加えて、冗長性無しのディスクアレイシステムをレベル0と呼ぶこともある。上記の各レベルは冗長性などにより実現するためのコストや性能特性などが異なるため、ディスクアレイシステムを構築するにあたって、複数のレベルのアレイ（ディスク装置の組）を混在させることも多い。ここでは、この組のことをパーティグループと呼ぶ。

【0003】ディスク装置は、性能や容量などによりコストが異なり、ディスクアレイシステムを構築するにあたって最適なコストパフォーマンスを実現するために、やはり性能や容量の異なる複数のディスク装置を用いることがある。

【0004】ディスクアレイシステムに格納されるデータを上記のようにディスク装置に分散して配置するため、ディスクアレイシステムは、ディスクアレイシステムに接続するホストコンピュータがアクセスする論理記憶領域とディスク装置の記憶領域を示す物理記憶領域の対応づけ（アドレス変換）を行う。特開9-27454号公報には、ホストコンピュータからの論理記憶領域に対する1/Oアクセスについての情報を取得する手段と、論理記憶領域の物理記憶領域への対応づけを変更して物理的再配置を行う手段により、格納されたデータ

の最速配置を実現するディスクアレイシステムが開示されている。

【0005】

【発明が解決しようとする課題】特開9-27454号公報に示されるような従来の技術における配置最適化の実行方法については以下の課題がある。

【0006】再配置する論理記憶領域の選択および再配置先の物理記憶領域の選択にあたり、ディスクアレイシステムのユーザまたは保守員が、前記ディスクアレイシステムの構成や個々のディスク装置の特性や性能などの情報を参照して前記選択を行わなければならない。ユーザまたは保守員による作業が煩雑となっていた。

【0007】また、ディスクアレイシステムが選択を自動的に行う場合においても、ユーザまたは保守員が前記個々のディスク装置の情報を参照して選択基準値を規定しなければならない。やがてユーザまたは保守員による作業が煩雑となっていた。特に、上記のように異なるレベルや異なるディスク装置の混在するディスクアレイシステムについては情報管理の煩雑さが増大する。

【0008】また、ディスクアレイシステムが選択のために1/Oアクセス情報の参照は、ホストコンピュータおよびディスクアレイシステムを含むシステムで行われる処理のスケジューリングの特性を考慮していない。従って、一般にコンピュータシステムで行われる処理と、それに伴う1/Oは、ユーザによって作成されたスケジューリングに則って行われており、また処理および1/Oの傾向は日毎、月毎、年毎などの周期性を示す場合も多く、一般にユーザは特定期間の処理および1/Oに関心があると考えられる。

【0009】上記従来技術において、再配置による性能チューニング方法については以下の課題がある。物理的再配置による性能チューニング方法は、ディスク装置、すなわち、物理記憶領域の使用状況に変更を加えるものであるが、従来の技術においては、ホストコンピュータからの論理記憶領域に対する1/Oアクセスについての情報を参照するため、再配置する論理記憶領域の選択および再配置先の物理記憶領域の選択にあたり、正しい選択が行えない可能性があった。

【0010】また、ホストコンピュータからのシーケンシャルアクセスとランダムアクセスが混在し、同一のディスク装置に含まれる別々の物理記憶領域に対して行われる場合でも、シーケンシャルアクセスとランダムアクセスを異なるディスク装置に分散するために、再配置先のディスク装置を任意に特定して自動的再配置を行わせることはできなかった。一般に、ホストコンピュータからの処理要件として、データ長の小さいランダムアクセスは短時間での応答（高応答性）が求められるが、同一ディスク装置にデータ長の大きいシーケンシャルアクセスが存在する場合、ランダムアクセスの応答時間はシーケンシャルアクセスの処理に阻害されて長くなり、

応答性能は悪化してしまう。

【0011】本発明の第一の目的は、ディスクアレイシステムのユーザまたは保守員が再配置による配置最適化を行うための作業を簡便にすることにある。

【0012】本発明の第二の目的は、ホストコンピュータおよびディスクアレイシステムを含むシステムでの処理のスケジューリングを考慮した再配置による配置最適化を可能にすることにある。

【0013】本発明の第三の目的は、再配置する論理記憶領域の選択および再配置先の物理記憶領域の選択にあたり、実際の記憶装置であるディスク装置の使用状況に基づき選択を行う、ディスクアレイシステムの制御方法およびディスクアレイシステムを提供することにある。

【0014】本発明の第四の目的は、ディスクアレイシステムにおける同一ディスク装置での異なるシーケンシャルアクセスとランダムアクセスの混在に対し、再配置先のディスク装置を任意に特定して再配置によりシーケンシャルアクセスおよびランダムアクセスを異なるディスク装置に自動的に分離することができるとすることにある。

【0015】

【課題を解決するための手段】上記の第一の目的を実現するために、1台以上のホストコンピュータに接続するディスクアレイシステムは、配下の複数のディスク装置の使用状況情報を取得する手段と、ホストコンピュータがリード/ライト対象とする論理記憶領域とディスク装置の第一の物理記憶領域との対応づけを行う手段とを有し、さらに、複数のディスク装置をそれぞれ属性を有する複数の組（クラス）として管理する手段と、使用状況情報およびクラス属性に基づき論理記憶領域に対する再配置先のクラス属性を決定する手段と、論理記憶領域の再配置先として利用可能な第二の物理記憶領域をクラス内から選択する手段と、第一の物理記憶領域の内容を前記第二の物理記憶領域にコピーするとともに論理記憶領域の対応づけを第一の物理記憶領域から第二の物理記憶領域へ変更して再配置を行う手段を備える。

【0016】また、上記第二の目的を実現するために、ディスクアレイシステムは、使用状況情報を蓄積し、設定された期間の使用状況情報に基づき、論理記憶領域の再配置先を決定する手段と、設定された時間に再配置を行う手段を備えることができる。

【0017】また、上記第三の目的を実現するために、ディスクアレイシステムは、使用状況情報として、ディスク装置の単位時間当たりの使用時間（使用率）を用いる手段を備える。

【0018】また、上記第四の目的を実現するために、ディスクアレイシステムは、各クラスに属性として設定された対象アクセス種別（シーケンシャル/ランダムアクセス種別）と使用率上限値を用いて、クラスの使用率上限値を超えている記憶装置から再配置する論理記憶領域

域を選択し、論理記憶領域に対するアクセス種別の分析結果に基づいて論理記憶領域の再配置先のクラスを最適なアクセス種別のクラスから、各クラスの使用率上限値を超えないように決定する手段を備える。

【0019】

【発明の実施の形態】以下、本発明の実施の形態を図1～図27を用いて説明する。

【0020】<第一の実施の形態>本実施の形態では、クラス600に基づき再配置の判断と、再配置判断および実行のスケジューリングについて説明する。

【0021】図1は、本発明の第一の実施の形態における計算機システムの構成図である。

【0022】本実施の形態における計算機システムは、ホスト100、ストレージサブシステム200、制御部150を有している。

【0023】ホスト100は、ストレージサブシステム200に1/Oバス800を介して接続し、ストレージサブシステム200に対しリード/ライトの1/Oを行う。1/Oの際、ホスト100は、ストレージサブシステム200の記憶領域について論理領域を指定する。1/Oバス800の例としては、ESCON、SCSI、ファイバチャネルなどがある。

【0024】ストレージサブシステム200は、制御部300および複数の記憶装置500を有する。制御部300は、リード/ライト処理310、使用状況情報取得処理311、再配置判断処理312、および再配置実行処理313を行う。また、ストレージサブシステム200は、論理/物理対応情報400、クラス構成情報401、クラス属性情報402、論理領域使用状況情報403、物理領域使用状況情報404、再配置判断対象期間情報405、再配置実行時刻情報406、未使用領域情報407、および再配置情報408を保持する。

【0025】ホスト100、制御部300、および制御部端末700は、ネットワーク900で接続される。ネットワーク900の例としては、FDDI、ファイバチャネルなどがある。

【0026】ホスト100、制御部300、および制御部端末700には、各々での処理を行うためのメモリ、CPUなど、計算機において一般に用いられる構成要素もそれぞれ存在するが、本実施の形態の説明においては重要でないため、ここでは説明を省略する。

【0027】ホスト100が、ストレージサブシステム200に対してリード/ライトを行う場合のリード/ライト処理310、および使用状況情報取得処理311について図2で説明する。

【0028】リード/ライト処理310において、ホスト100は、ストレージサブシステム200の制御部300に対しリードまたはライトを論理領域を指定して要求する（ステップ1000）。要求を受領した制御部300は、論理/物理対応情報400を用いて論理領域に

対応する物理領域を求め、すなわち論理領域のアドレス（論理アドレス）を物理領域のアドレス（物理アドレス）に変換する（ステップ1010）。続いて制御部300は、リードの場合は、この物理アドレスの記憶装置500からデータを読み出してホスト100に転送し、ライトの場合は、ホスト100から転送されたデータを前記物理アドレスの記憶装置500に格納した（ステップ1020）。さらに後述の使用状況情報取得処理311を行う。リード/ライト要求およびデータ転送は1/Oバス800を介して行われる。

【0029】論理/物理対応情報400の一例を図3に示す。論理アドレスはホスト100がリード/ライト処理310で用いる論理領域を示すアドレスである。物理アドレスは実際にデータが格納される記憶装置500上の領域を示すアドレスであり、記憶装置番号および記憶装置内アドレスからなる。記憶装置番号は個々の記憶装置500を示す。記憶装置内アドレスは記憶装置500内の記憶領域を示すアドレスである。

【0030】次に、使用状況情報取得処理311において制御部300は、リード/ライト処理310においてリード/ライト対象となった論理領域についての論理領域使用状況情報403と、リード/ライト処理310で使用した物理領域についての物理領域使用状況情報404を更新する（ステップ1030、1040）。論理領域使用状況情報403および物理領域使用状況情報404は、例えば使用頻度、使用率、リード/ライトに関する属性など、各々の論理領域と物理領域の各日時の使用状況に関する情報である。論理領域使用状況情報403および物理領域使用状況情報404の具体的な例は、以降の実施形態で説明する。

【0031】次に、制御部300が行う再配置判断処理312について図4で説明する。

【0032】記憶装置500は、ユーザによって、または初期状態として複数の組（クラス600）に分類されており、クラス600への分類はクラス構成情報401によって設定されている。さらに、各クラス600は、ユーザによって、または初期条件として属性を設定されており、属性は、クラス属性情報402に設定されている。クラス属性情報402は、許容使用状況や好適な使用状況やクラス間優先順位などの属性に関する情報である。クラス構成情報401およびクラス属性情報402の具体的な例は、以降の実施形態で説明する。再配置判断対象期間情報405には、ユーザによってまたは初期条件として再配置判断処理312の対象とする使用状況情報の期間と期間更新情報406が設定されている。

【0033】再配置判断対象期間情報405の一例を図5に示す。開始日時から終了日時までの期間が対象期間となる。期間更新情報は今回の対象期間の設定条件であり、例えば毎週、毎日、X時間後などがありうる。制御部300は、対象期間の論理領域使用状況情報403お

よび物理領域使用状況情報404を参照し（ステップ1100）、クラス属性情報402の各クラス600の許容使用状況などと比較して（ステップ1110）、物理的再配置を行うべき論理領域を選択する（ステップ1120）。

【0034】さらに、制御部300は、クラス属性情報402の許容使用状況や好適な使用状況やクラス間優先順位などを参照して（ステップ1130）、論理領域の再配置先のクラス600を選択し（ステップ1140）、さらに、クラス600に属する記憶装置500の中から論理領域の再配置先として未使用の物理領域を選択し（ステップ1150）、選択結果を再配置情報408に出力する（ステップ1160）。

【0035】再配置情報408の一例を図6に示す。論理領域は、再配置する論理領域であり、再配置元物理領域は、論理領域に対応する現在の物理領域を示す記憶装置番号と記憶装置内アドレスであり、再配置先物理領域は、再配置先の物理領域を示す記憶装置番号と記憶装置内アドレスである。図6に示すように再配置の立案は一つ以上行われる。さらに制御部300は、再配置判断対象期間情報405の期間更新情報を参照して、再配置判断対象期間情報405の対象期間を次回分更新する（ステップ1170）。上記の処理において制御部300は、論理/物理対応情報400を用い、また前記の未使用の物理領域の検索に未使用領域情報407を用いる。

【0036】未使用領域情報407の一例を図7に示す。記憶装置番号は個々の記憶装置500を示す。記憶装置内アドレスは記憶装置500内での領域を示すアドレスである。記憶装置番号および記憶装置内アドレスは物理領域を示し、使用/未使用の項目は、物理領域の使用/未使用の区別を示す。制御部300は、通常、再配置判断処理312を対象期間以後、後述の再配置実行処理313以前に自動的に行う。

【0037】次に、制御部300が行う再配置実行処理313について図8で説明する。

【0038】再配置実行時刻情報406にはユーザによってまたは初期条件として再配置実行処理313を行う日時と日時更新情報406が設定されている。

【0039】再配置実行時刻情報406の一例を図9に示す。制御部300は、設定された日時以下に説明する再配置実行処理313を自動的に実行する。日時更新情報は今回の再配置実行処理313を行う日時の設定条件であり、例えば毎週、毎日、X時間後などがありうる。

【0040】再配置情報408に基づいて再配置元物理領域に格納している内容を再配置先物理領域にコピーする（ステップ1200）。さらに、コピーが完了して再配置元物理領域の内容が全て再配置先物理領域に反映された時点で、制御部300は、論理/物理対応情報400上の再配置を行う論理領域に対応する物理領域

を再配置元物理領域から再配置先物理領域に変更する（ステップ1210）。

【0040】さらに、制御部300は、未使用物理領域4700上の再配置元物理領域を使用し、再配置元物理領域を未使用に変更する（ステップ1220）。さらに、制御部300は、再配置実行時刻情報406の日時更新情報を参照して、再配置実行時刻情報406の日時を次回分更新する（ステップ1230）。

【0041】ユーザまたは保守員は、制御部300が上記の処理で用いている各情報を、制御部700からホストワーク900を介して、またはホスト100からホストワーク900または1/Oバス800を介して設定および確認すること、特に、再配置情報408を確認および設定して再配置案を修正や追加や削除などを行うことができる。

【0042】上記の処理を行うことにより、取得した使用状況情報および設定されたクラス属性に基づいて、ストレージサブシステム200において論理領域の物理的再配置を自動的に実行し、ストレージサブシステム200の最適化を行うことができる。さらに上記の再配置判断および実行の処理を繰り返して配置を修正していくことにより、使用状況の変動やその他の最適化要因を吸収していくことができる。

【0043】特に、上記の処理により、ユーザまたは保守員は再配置による最適化を簡単に行うことができる。ユーザまたは保守員は、記憶装置500をクラス600という単位で管理できるため、記憶装置500の性能や信頼性や特性などの属性を個々の記憶装置500について管理する必要がある。さらに、ユーザまたは保守員は、記憶装置500の個々の属性が等しくない組に対して、必要に応じて同一の属性を持つクラス600を設定して、1つの管理単位として扱うことができる。ただし、1つの記憶装置500が1つのクラス600を構成すると見なして1つの記憶装置500を管理単位として上記の再配置の処理を行うことも可能である。

【0044】また、ユーザまたは保守員は、ホスト100で行われる処理（ジョブ）の特性やスケジュールを考慮して、上記の再配置を自動的に行うことができる。一般に、計算機システムで行われる処理と、この処理に伴う1/Oは、ユーザによって作成されたスケジュールに則って行われる。ユーザは、特に最適化の対象としたい処理であり、本実施形態で説明した再配置の処理が可能で、ユーザは関心のある期間を設定して再配置判断の処理をストレージサブシステム200に行わせ、すなわち、前記期間の使用状況情報に基づいて上記の再配置による最適化を実現することができる。また、計算機システムで行われる処理および1/Oの傾向は日毎、月毎、年毎などの周期性を示す場合も多い。特に、処理が定型業務に基づく処理である場合には、周期性が顕著となる。前述の場

合と同様にユーザは、周期において特に最適化対象として関心のある期間を指定して再配置による最適化を行うことができる。また、再配置実行処理313では、ストレージサブシステム200内で格納内容のコピーを行う。ユーザはストレージサブシステム200があまり使用されていない時刻やホスト100で実行されている処理の要求処理性能が低い期間を再配置実行処理313の実行時刻として設定することで、ホスト100での要求処理性能が高い処理のストレージサブシステム200への1/Oがコピーにより阻害されることを回避できる。

【0045】なお、記憶装置500は、それぞれ異なる性能、信頼性、特性や属性を持っていてよく、特に具体的に、磁気ディスク装置、磁気テープ装置、半導体メモリ（キャッシュ）のように異なる記憶媒体であってもよい。また、上記の例では未使用領域情報407は物理領域に基づいて記述されているとしたが、未使用の物理領域に対応する論理領域（論理アドレス）に基づいて記述されていてもよい。

【0046】＜第二の実施形態＞本実施形態では、使用状況情報としてのディスク装置使用率の適用と、クラス600の上限値およびクラス600間の性能順位による再配置判断について説明する。

【0047】図10は、本発明の第二の実施形態における計算機システムの構成図である。

【0048】本実施形態の計算機システムは、ホスト100、ディスクアレイシステム201、制御部700を有している。本実施形態における計算機システムは、第1の実施形態でのストレージサブシステム200をディスクアレイシステム201とし、記憶装置500をパーティティグループ501としたものに相当する。

【0049】ディスクアレイシステム201は、制御部300とディスク装置502を有する。制御部300は、第1の実施形態での制御部300に相当する。ディスク装置502は、n台（nは2以上の整数）でRAID（ディスクアレイ）を構成しており、このn台のディスク装置502による組をパーティティグループ501と呼ぶ。RAIDの性質として、1つのパーティティグループ501に含まれるn台のディスク装置502は、n-1台のディスク装置502の格納内容から生成される冗長データが壊れた1台に格納されるといった冗長性上の関係を有する。この関係から各パーティティグループ501は、パーティティグループ502に分け格納されるなど、データ格納上の関係を有する。この関係から各パーティティグループ501は、動作上の1単位とみなすことができるが、冗長性や台数nなどにより実現するためのコストや性能特性など異なるため、ディスクアレイシステム201を構成するにあたって、レベルや台数nの異なるアレイ（パーティティグループ501）を混在させることも多く、またパーティティグループ501を構成するディスク装置502につ

1. 固定性が設定されているクラス600またはパリディ
 ループ501を扱うことによって、ユーザは上記の自動
 的な再配置処理において物理的な再配置の影響を生じさ
 せたくないクラス600またはパリディグループ501
 を設定し、再配置の対象外とすることができる。

【0071】＜第3の実施の形態＞本実施の形態では、同一クラス600内で再配置判断について説明する。本実施の形態での計算ユニットは、第2の実施の形態と同様である。ただし、本実施の形態では1つのクラス600に複数のバリエーショングループ501が属する。本実施の形態での処理は、再配置判断処理312を除いては第2の実施の形態と同様である。また、再配置判断処理312についても、再配置する論理領域の選択（ステップ1600）は、第2の実施の形態と同様である。

【0072】本実施の形態での再配置判断処理312における、再配置先の物理領域の選択について図20で説明する。

【0073】第2の形態は再配置先の物理領域を再配置元の物理領域の属するクラス600より性能順に上位が低位のクラス600から選択するが、本実施の形態では同一クラス600の再配置元以外のパリティグループ501から選択する。制御部300は、クラス構成情報401と未使用領域情報407を参照して、同一クラス600に属する再配置元以外のパリティグループ501の未使用物理領域を算出する(ステップ1610)。

置換制御部 3 0 0 は、各未使用物理領域について、再配置先（ステップ 1 6 2 0）、未使用物理領域の中から、再配置先とした場合に同一クラス 6 0 0 に設定されている上限値を超えないと予測できる未使用物理領域を、再配置先の物理領域として選択し（ステップ 1 6 3 0）、選択結果を第 2 の実施形態図 4 0 8 に出方する（ステップ 1 6 4 0）。再配置する全ての論理領域について再配置先の物理領域を選択し終えたら処理を終了する（ステップ 1 6 5 0）。

【0074】上記の処理により、同一クラス600内に、上記の処理502の負荷を分散することができ、上記の処理502はディスタンスレイシステム200の1のパーティティグループ501が全て1つのクラス600（単一クラス）に属する情に適用することができ、また、例えば、第2の実施の形態で説明した処理方法と組み合わせると、所定優先の未使用物理領域の選択において、所定数のクラス600より性能順位が低かった場合や、性能順位が最上位のクラス600での処理に適用できず、第2の実施の形態で説明した処理方法と組み合わせると、第2の実施の形態での処理方法と本実施の形態での処理方法とを、各クラス600について異なる使用上限値を用いてもよく、すなわち、そのため、例えば、図402が各クラス600について

二種類の使用率上限値または差分を有してもよい。

【0075】＜第四の実施の形態＞本実施の形態では、第2の実施の形態との再配置判断処理312において、再配置元のクラス600より性能順位が高位のクラス600（高性能クラス）に再配置先の未使用領域が見つからなかった場合に、再配置先を得るために先行して行われる。性能順位がより低位のクラス600（低性能クラス）への高性能クラスからの再配置の処理について説明する。

【0076】本実施の形態での計算機システムは、第2の実施の形態と同様である。本実施の形態における再配置判断処理312について図21で説明する。

【0077】制御部300は、高性能クラスに属するパ
リティグループ501をクラス構成情報401から取得

する（ステップ1700）。続いて制御部300は、第1の実施の形態と同様の再配置判断対象期間情報405を参照して対象期間を取得し（ステップ1710）、対象期間の論理領域使用状況情報403を参照して、バ

ディグループ501の各物理領域に対応する論理領域の

ディस्क装置使用率を取得し(ステップ1720)、ディस्क装置使用率の小さいものから、低性能クラスへ再配置する論理領域として選択する(ステップ1730)。このとき論理領域の選択は必要だけ行われる(ステップ1740)。

【0078】続いて制御部300は、選択された論理傾

処理先の物理領域選択の処理は、第2の実施の形態での処理説明において再配置先としている高性能クラスを低性能クラスと読み替えば、第2の実施の形態での処理と同様である(ステップ1750)。また、本実施の形態におけるその他の処理も第2の実施の形態での処理と同様である。

【0079】上記の処理を行うことで、第2の実施の形態で配置判断処理312において高性能クラスに再配置先の未使用物理領域が見つからなかった場合に、高性能クラスから低性能クラスへ論理領域の再配置を、高性能クラスへの再配置に先行して行い、再配置先の未使用物理領域を高性能クラスに必要とすることができる。制御部300は、上記の処理を必要に応じて繰り返して行い、十分な未使用物理領域を用意することができる。

【0080】論理領域の再配置先を低性能クラスのパーティティグループ501とするため、同一負荷に対するディクテーションの使用時間が再配置について増大し、論理領域の再配置後のディスクアクセス装置使用率が増大する可能性があるが、ディスク使用率の小さい論理領域から再配置していくことで、増大の影響を最小限に抑えることができる。

【0081】＜第五の実施の形態＞本実施の形態では、クラス600の属性の1つにアクセス種別属性を設け、

アグセス感別属性を用いてシーケンシャルアクセスが顕著に行われる論理領域とランダムアクセスが顕著に行われる論理領域とを、他のバリエーション501に自動的に物理的再配置して分離するための再配置判断について説明する。

【0082】本実施の形態における計算機システムは図10に示したものである。本実施の形態では、第2の実施の形態での説明に加え、制御部300が保持する下記の情報を用いる。

【0083】本実施の形態でのクラス属性情報402の一例を図22に示す。この例では、第2の実施の形態で例に対してアクセス種別が加えられており、クラス600のアクセス種別が、例えばシーケンシャルに設定されている場合は、クラス600がシーケンシャルアクセスに好適であると設定されていることを示す。

【0084】本実施の形態での論理領域使用状況情報4 03の一例を図23に示す。この例では、第2の実施の形態での例に対し、シーケンシャルアクセス率およびランダムアクセス率が加えられている。

【0085】さらに、本実施の形態において制御部300は、第2の実施の形態に加え、アクセス種別基準値情報410と論理領域属性情報411を保持する。

【0086】アクセス種別基準値情報410の一例を図24に示す。ユーザによりまたは初期条件としてアク

セグメント基礎価値情報 410には後述のアクセス権別と性別別の基礎価値が設定されている。また、論理領域風定に用いる基準値が設定されている。図 25 に示す、アクセス権別と性別情報 411 の一例を図 26 に示す。アクセス権別については、各論理領域について顧客に行われると期待できるアクセス権別であり、ユーザが設定する。固定については後述する。

【0087】本実施の形態では、使用状況情報取得処理311および再配置判断処理312を除いては第一の実施の形態と同様である。

【0088】本実施の形態における使用状況情報取得処理311について図26で説明する。

【0089】制海部300は、第2の実施の形態での使用状況情報取得処理311と同様に、隣接領域についてのディスク装置使用率を算出し（ステップ1800、1810）、リード/ライト処理310での使用率内容を分析して、使用率についてシーケンシャルアクセスとランダムアクセスの比率を算出し（ステップ1820）、

使用率およびアクセス種別比率を論理領域使用状況情報
404に記録する(ステップ1830)。また、制御部
300は、第2の実施の形態と同様にパリティグループ
の使用率の算出と物理領域使用状況情報404への記録を
行う(ステップ1840、1850)。

【0090】本実施の形態における再配置判断処理312において、再配置する論理領域の選択は第2の実施の形態と同様である（ステップ1990）。再配置判断処理312での再配置先の物理領域の選択について図27

で説明する。

【0091】制御部300は、論理領域使用情報403を参照し、再配置する論理領域についてのシーケンシャルアクセス事象を取得し(ステップ1910)、アクセス

種別基準値情報410に設定されている基準値と比較する(ステップ1920)。シークンシャルアクセス率がある基準値より大きい場合、制御部300は、クラス風性情報402を参照し、アクセス種別がシークンシャルと設定されているクラス600(シークンシャルクラス)が存在するか調べる(ステップ1950)。シークンシャルクラスが存在する場合、制御部300は、クラス構成情報401と未使用領域情報407を参照して、シークンシャルクラスに属する配置アドレス以外のパーティション501の未使用物理領域を参照する(ステップ1960)。さらに制御部300は、各未使用物理領域について

て、再配置先とした場合のパリティグループ使用率の予測値を求め（ステップ1970）、未使用物理領域の中から、再配置先とした場合にシークンシャルクラスに設定されている上限値を超えないと予測できる未使用物理

領域を、再配置先の物理領域として選択（ステップ1980）、選択結果第2の実施の形態と同様に再配置情報408に出力する（ステップ1990）。制御部300は、使用率予測値を、第2の実施の形態と同様のパリティ情報409と本実施の形態における論理領域使用状況情報403および物理領域使用状況情報404から算出する。

【0092】前記の比較において、シーケンシャルアクセス率が基準値以下である場合、制御部300は、論理領域性情報411を参照し、論理領域についてアクセス頻別ヒントがシーケンシャルと設定されているか調べ

る(ステップ1940)。アクセス規則にヒントにシーク
 ジャンルと設定されていた場合、上記1回以上制御部3
 000は、シークジャンルクラスの有無を調べ(ステップ
 1950)、シークジャンルクラスが存在する場合は、
 シークジャンルクラスから再選択先の物理頭像を選択す
 る(ステップ1960~1990)。

【0093】前記の比較において、シーケンシャルアクセス事が前記基準値以下であり、さらにアクセス頻度がシーケンシャルでなかった場合、またはシーケンスがシーケンシャルでなかった場合、制御部300は、第2の実地の形態と同様に、シーケンシャルクラス以外のクラス600から所配優先の物理領域を選択する(ステップ2000)。

【0094】上記の処理により、同一パーティティグループ501での顕著なシグネチャルアクセスとランダムアクセスの混在に対し、各クラス600に属性として設定されたアクセス頻度と使用率上限値を用いて、シグネチャルアクセスが顕著に行われる論理領域とランダムアクセスが顕著に行われる論理領域とを、異なるパーティティグループ501に自動的に面配列して分けて管理される。

なるディスク装置502に分離することができ、特にランダムアクセスに対する応答性を改善することができる。

【0095】また、上記の処理においては制御部300は、シーケンシャルアクセスに注目して再配置による自動的分離を行うとしたが、同様にランダムアクセスに注目して前記分離を行うことも可能である。

【0096】上記の再配置処理312において、再配置する論理領域を選択した時点で、制御部300が論理領域411を参照し、論理領域に固定属性が指定されている場合は、論理領域を再配置しないとする。ユーザが特に再配置を行いたくないと考える論理領域がある場合、固定属性を指定することで論理領域を再配置の対象外とすることができ、上記の固定属性に関する処理は論理領域固有属性情報411を用いることで、前述の実施の形態にも適用できる。

【0097】

【発明の効果】ストレージサブシステムのユーザ、または保守員が、記憶領域の物理的再配置による配置最適化を行うための作業を簡便にすることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態での計算機システムの構成図である。

【図2】本発明の第1の実施の形態でのリード/ライト処理310および使用状況情報取得処理311のフローチャートである。

【図3】本発明の第1の実施の形態での論理/物理対応情報400の一例を示す図である。

【図4】本発明の第1の実施の形態での再配置判断処理312のフローチャートである。

【図5】本発明の第1の実施の形態での再配置判断対象期間情報405の一例を示す図である。

【図6】本発明の第1の実施の形態での再配置情報8の一例を示す図である。

【図7】本発明の第1の実施の形態での未使用領域情報407の一例を示す図である。

【図8】本発明の第1の実施の形態での再配置実行処理313のフローチャートである。

【図9】本発明の第1の実施の形態での再配置実行時刻情報406の一例を示す図である。

【図10】本発明の第2の実施の形態および第5の実施の形態の計算機システムの構成図である。

【図11】本発明の第2の実施の形態での論理/物理対応情報400の一例を示す図である。

【図12】本発明の第2の実施の形態でのクラス構成情報401の一例を示す図である。

【図13】本発明の第2の実施の形態でのクラス属性情報402の一例を示す図である。

【図14】本発明の第2の実施の形態での使用状況情報取得処理311のフローチャートである。

【図15】本発明の第2の実施の形態での論理領域使用状況情報403の一例を示す図である。

【図16】本発明の第2の実施の形態での物理領域使用状況情報404の一例を示す図である。

【図17】本発明の第2の実施の形態での再配置判断処理312のフローチャートである。

【図18】本発明の第2の実施の形態でのパリティグループ情報409の一例を示す図である。

【図19】本発明の第2の実施の形態での再配置実行処理313のフローチャートである。

【図20】本発明の第3の実施の形態での再配置判断処理312のフローチャートである。

【図21】本発明の第4の実施の形態での再配置判断処理312のフローチャートである。

【図22】本発明の第5の実施の形態でのクラス属性情報402の一例を示す図である。

【図23】本発明の第5の実施の形態での論理領域使用状況情報403の一例を示す図である。

【図24】本発明の第5の実施の形態でのアクセス頻別基準値情報410の一例を示す図である。

【図25】本発明の第5の実施の形態での論理領域属性情報411の一例を示す図である。

【図26】本発明の第5の実施の形態での使用状況情報取得処理311のフローチャートである。

【図27】本発明の第5の実施の形態での再配置判断処理312のフローチャートである。

【符号の説明】

100 ホスト

200 ストレージサブシステム

201 ディスクアレイシステム

300 制御部

310 リード/ライト処理

311 使用状況情報取得処理

312 再配置判断処理

313 再配置実行処理

400 論理/物理対応情報

401 クラス構成情報

402 クラス属性情報

403 論理領域使用状況情報

404 物理領域使用状況情報

405 再配置判断対象期間情報

406 再配置実行時刻情報

407 未使用領域情報

408 再配置情報

409 パリティグループ情報

410 アクセス頻別基準値情報

411 論理領域属性情報

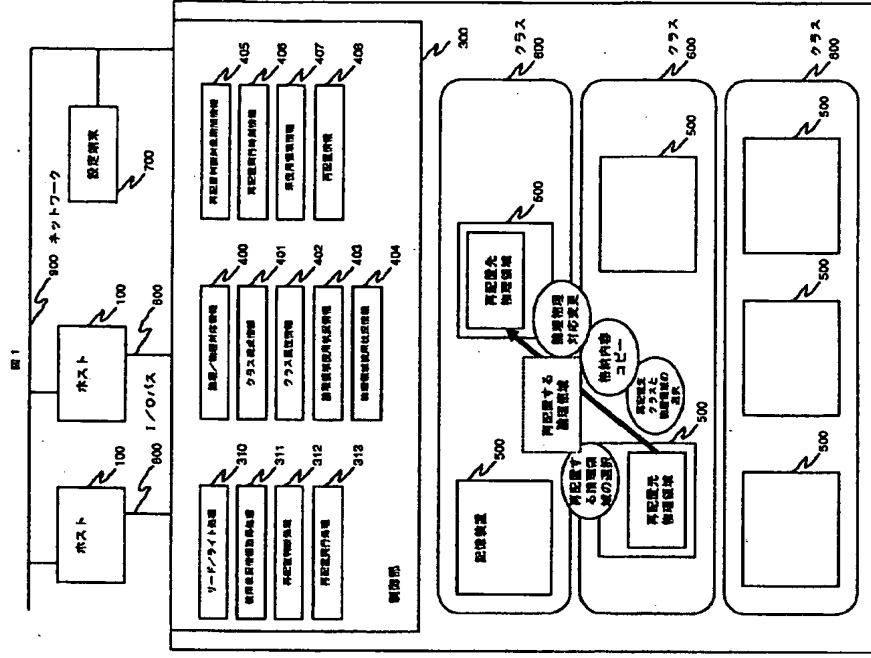
500 記憶装置

501 パリティグループ

502 ディスク装置

600 クラス
700 制御端末
800 I/Oバス
900 ネットワーク

【図1】



【図9】

【図24】

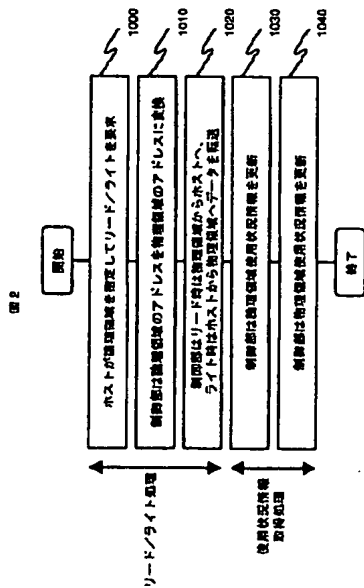
図9

図24

日時	1999年8月11日 22時0分
日時更新情報	毎日 (+2.4時間)

アクセス頻別基準値 (%)	75
---------------	----

【図2】



【図3】

図3

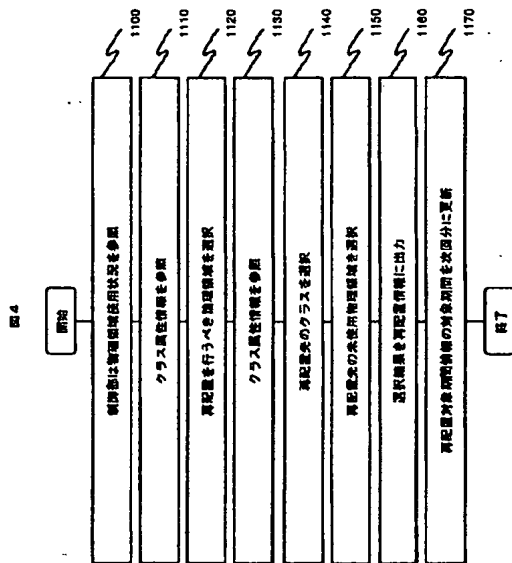
記憶アドレス	記憶アドレス	
	記憶部番号	記憶部内アドレス
0~999	0	0~999
1000~1999	0	1000~1999
2000~2999	1	0~999
3000~3999	1	1000~1999

【図5】

図5

開始日時	1999年8月11日 8時30分
終了日時	1999年8月11日 17時15分
所要時間	曜日 (+2.4時間)

【図4】



【図6】

図6

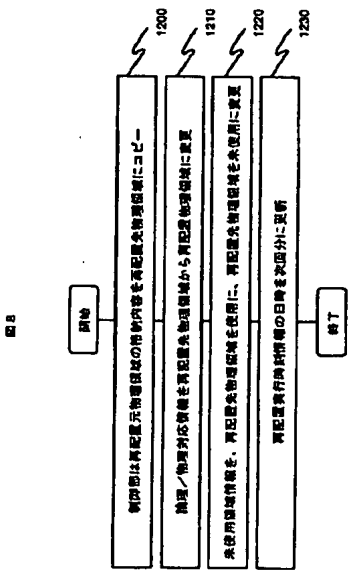
番号	記憶部	再配置先記憶部		再配置先記憶部	
		記憶部番号	記憶部内アドレス	記憶部番号	記憶部内アドレス
1	0~999	0	0~999	10	0~999
2	1000~1999	0	1000~1999	10	1000~1999

【図7】

図7

記憶部番号	記憶部内アドレス	使用/未使用
0	0~999	使用
0	1000~1999	使用
0	2000~2999	未使用
0	3000~3999	未使用

【図8】



【図11】

図11

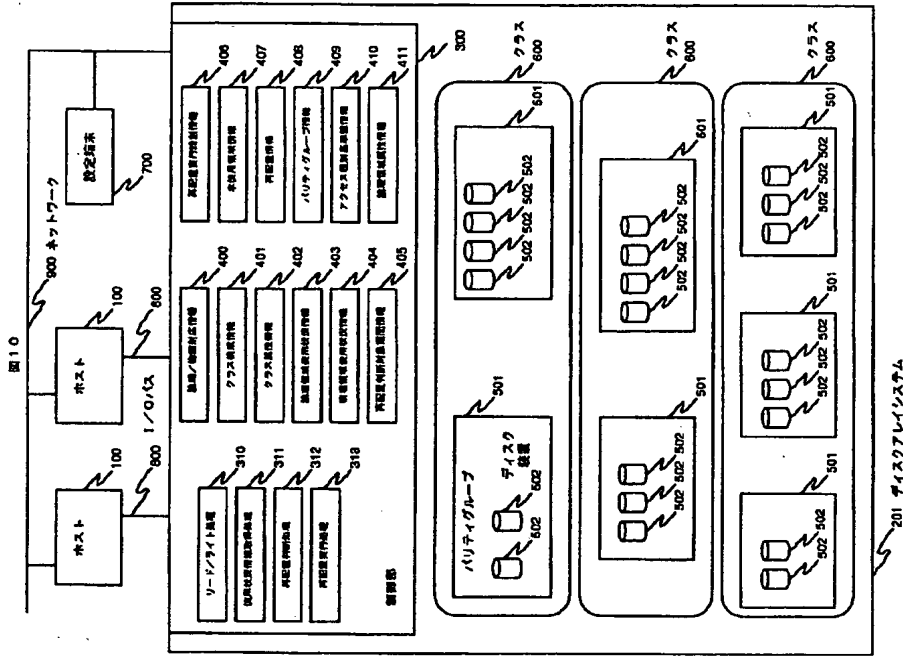
物理アドレス	物理アドレス			更新データ	
	パリティグループ番号	記憶装置番号	記憶装置内アドレス	記憶装置番号	記憶装置内アドレス
0-999	100	0	0-999	20	0-999
1000-1999	100	0	1000-1999	20	1000-1999
2000-2999	101	1	0-999	41	0-999
3000-3999	101	1	1000-1999	41	1000-1999

【図12】

図12

クラス番号	パリティグループ番号	パリティグループ番号
0	9	100, 110, 120
1	2	101, 111
2	4	102, 112, 122, 132

【図10】



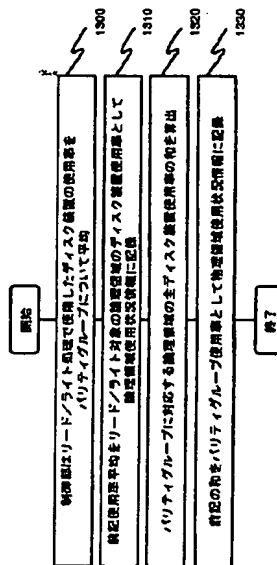
【図13】

図13

クラス番号	使用率上限値 (%)	クラス間移動位置	再配置実行上限値 (%)	固定
0	60	1	70	-
1	70	2	80	固定
2	80	3	90	-

【図14】

図14



【図15】

図15

日時	固定アドレス	ディスク管理使用率 (%)
1999年8月11日 8時0分	0~999	18
	1000~1999	32
1999年8月11日 8時15分	0~999	20
	1000~1999	30
1999年8月11日 8時30分	0~999	22
	1000~1999	28

【図16】

図16

日時	パリティグループ番号	使用率 (%)
1999年8月11日 8時0分	100	68
	101	92
1999年8月11日 8時15分	100	70
	101	80
1999年8月11日 8時30分	100	72
	101	48

【図18】

図18

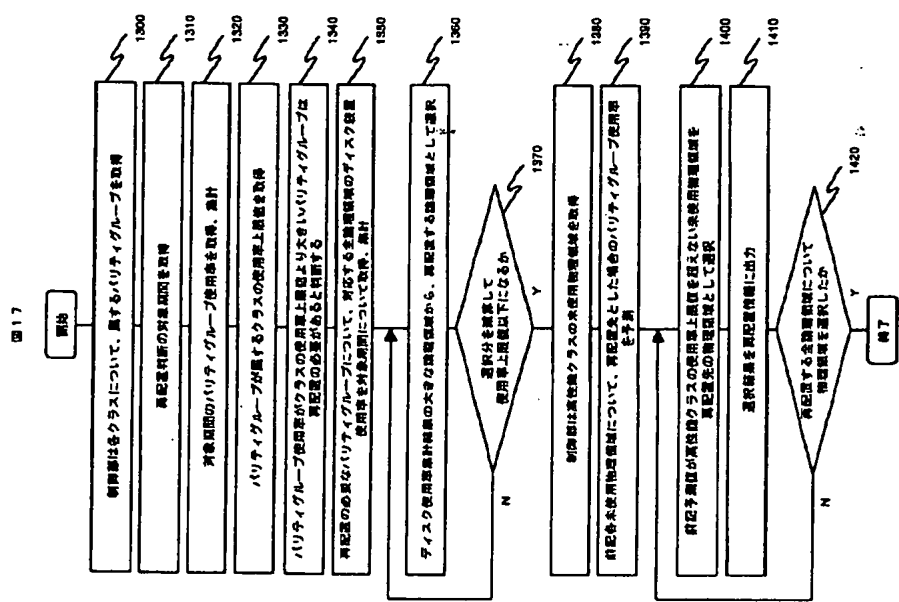
パリティグループ番号	RAID構成	ディスク組数	固定
100	RAID5 3D1P	110	-
101	RAID1 1D1P	100	固定
102	RAID5 6D1P	99	-

【図22】

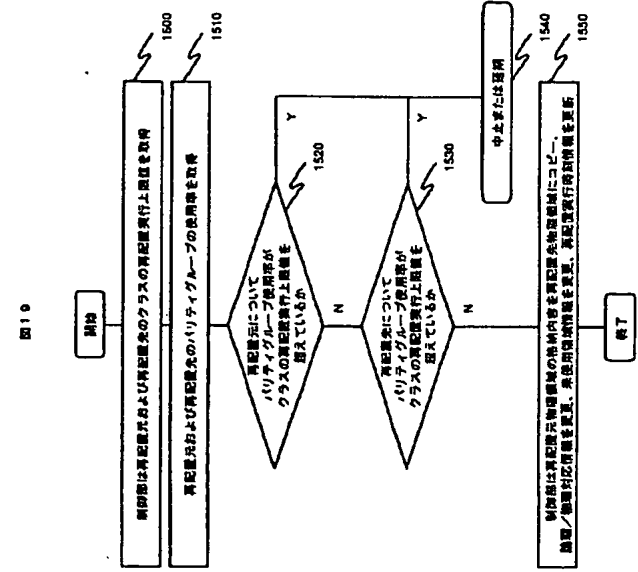
図22

クラス番号	使用率上限値 (%)	クラス間移動位置	再配置実行上限値 (%)	固定	アクセス制御
0	60	1	70	-	-
1	70	2	80	-	-
2	80	3	90	-	シーケンシャル

【図17】



【図19】

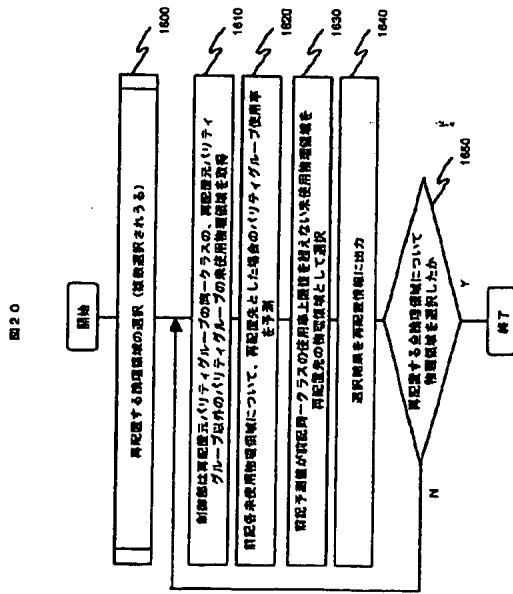


【図23】

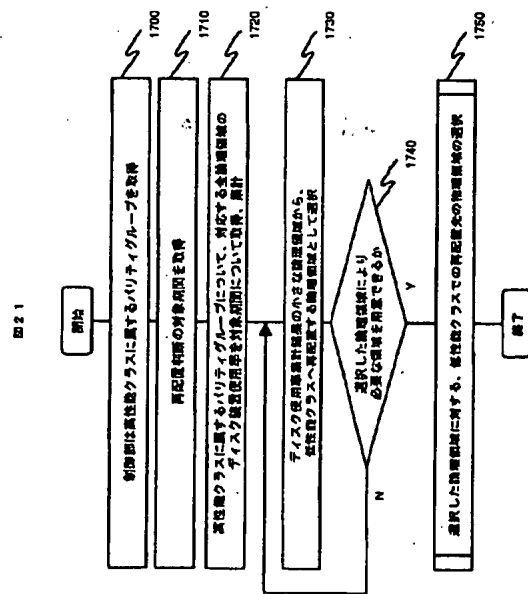
図23

日時	経過アドレス	ディスク使用率 (%)	ランダムアクセス率 (%)	ランダムアクセス率 (%)
1999年8月11日 8時0分	0~999	18	78	22
	1000~1999	32	92	48
1999年8月11日 8時15分	0~999	20	80	20
	1000~1999	30	50	50
1999年8月11日 8時30分	0~999	22	82	18
	1000~1999	28	48	82

【図20】



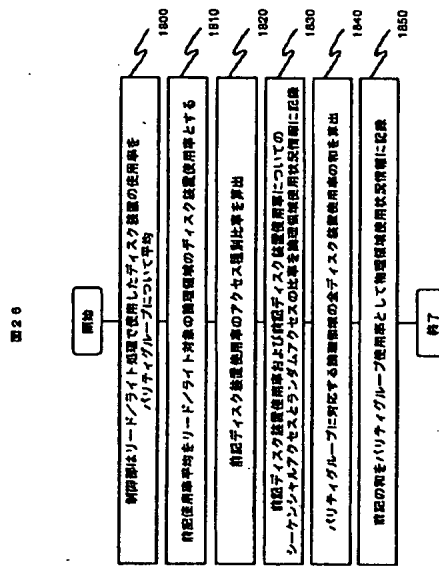
【21】



【图25】

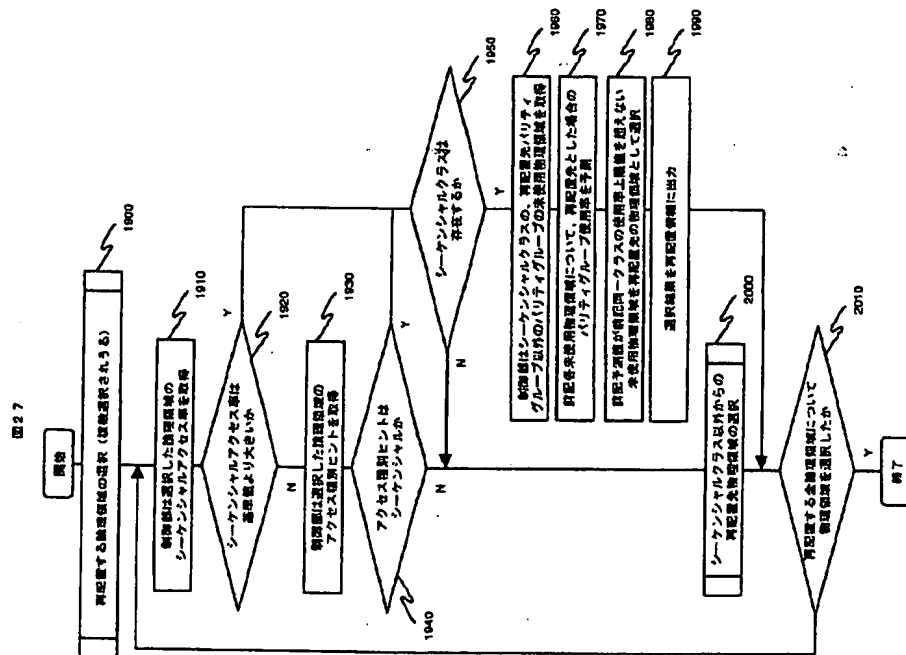
陸軍アドレス	アクセス数にヒント	固定
0～999	—	—
1000～1999	—	—
2000～2999	シーケンシャル	—
3000～3999	—	固定

【图26】



Fターム(参考) 5B065 BA01 CA30 CC01 CC03 EX01
5B082 CA11

【图27】



フロントページの続き

(72)発明者 山神 聡司
神奈川県川崎市麻生区王碑寺1099番地 株式会社日立製作所システム開発研究所内

(72)発明者 荒井 弘治
神奈川県小田原市国府津280番地 株式会社日立製作所ストレージシステム事業部内

高機能ディスクにおけるアクセスプランを用いた プリフェッチ機構に関する評価

向井 景洋, 根本 利弘, 喜連川 俊

東京大学 生産技術研究所

〒106-8558 東京都港区六本木 7-22-1

Tel 03-3402-6231 Fax 03-3479-1706

E-mail: {mukai,nemoto,kitsure}@tkl.iis.u-tokyo.ac.jp

概要

半導体技術の進歩により、プロセッサ、メモリは急速に高性能化しているにも関わらず、コストは急激に下がってきている。このような状況を背景に、出なる容量としてのディスクではなく、高性能のコンローラや大容量のメモリを搭載し、複雑な処理を行うことの出るディスクに注目が集まっている。現在、様々な研究機関で、インテリジェントなディスクに関する研究が行われている。これらの研究で提案されているディスクは、データベースアプリケーションに関する研究が向上させる一方で、ホストのコードの大幅な変更を必要とする。そこで、本論文では、アプリケーションレベルの知識を理解し、効率的なI/O処理を行う、インテリジェントなディスクを提案する。このようなディスクでは、ホストコードの変更が少ないことが特徴として挙げられる。このディスクに於いては、アクセスプランを用いたプリフェッチを行った場合に、どの程度の性能向上が見込まれるかを調べるために、シミュレーション実験を行った。その結果、実行時間の大幅な削減が期待できることが判った。

Evaluation of Prefetching Mechanism Using Access Plan on Intelligent Disk

Kagehiro Mukai, Toshihiro Nemoto and Masaru Kitsuregawa

Institute of Industrial Science, The University of Tokyo

7-22-1, Roppongi, Minato-ku, Tokyo 106-8558, Japan

Abstract

The progress of semiconductor technologies has made high performance processors and memory chips available at cheaper price. Therefore, it has become possible for disks to process complicated tasks. There are several research projects in the field of intelligent disk architecture. Although they improve the performance of database applications significantly, dbms code has to be modified a lot. In this paper, we examine the access plan based prefetching approach where the access plan is given to the disk controller. This approach needs very little change in host code. We examine how our access plan based prefetching approach improves the performance in detail. Experiment results indicate that the execution time decreases dramatically.

1 はじめに

半導体技術の進歩により、プロセッサ、メモリは急速に高性能化しているにも関わらず、そのコストは急激に下がってきている。現在のハイエンドなディスクのコンローラには100MHzのプロセッサが用いられ、ディスク・キャッシュのサイズはすでに4MB(最大16MB)[8]に達している。そのため、ロバットアームなどの物理的な動きを必要とするシーク時間や回転待ち時間が数ミリ秒であるのに対して、0.1ミリ秒のオーダーで処理を終えるコンローラの性能が顕著に上がっている。一方、データベースの世界では、DSSやデータ・ウェアハウスは急速に大規模化し、システムにおける処理性能の向上や容量の増加が強く要求されている。データの急速な増加に対応するために、ユーザは膨大な量のデータを複数のディスクに保存する。大量なデータを多数のディスクに保持し、並列にアクセスするシステムにおいては、ホストのコンピュータはディスク間の調整に大部分の時間を割くことが必要となる。また、ホストとディスクを繋ぐネットワークのバンド幅はボトルネックになりやすい。

以上で述べた事を背景に、単なる容量としてのディスクではなく、高性能のコンローラや大容量のメモリを搭載し、複雑な処理を行うことの出るインテリジェントなディスクに注目が集まっている。

本論文では、アプリケーションレベルの知識を理解し、それを基に効率的なI/O処理を行うインテリジェントなディスクを提案する。このようなディスクでは、アプリケーションレベルの知識を理解できる為、負荷が比較的軽い時間を利用して、将来、ホストが参照すると予想されるデータを効率的な良いスケジューリングのもとでプリフェッチを行っておくことにより、キャッシュヒット率を向上させることができ、性能の向上が期待できる。

インテリジェントなディスクに関する研究例としては、データベースにおけるSELECT、GROUP BY、外部ソート、データキューブ、画像処理などのアプリケーションに於いて、ディスクに内在したストリームベータでデータを処理するディスクレットを用いた研究[3]、大規模なSNIPシステムにおいて、スケッチ、ハッシュジョイン、ソートとディスクで行う事により、ホストへのデータ転送を削減する研究[4]、近似データサーチ、データマイニング、画像処理のスキームベースのアルゴリズムを取った研究[2]、などが提案されている。これらの研究では、複雑な処理をディスク内で行う為、ホストのコードの大幅な変更を必要とする。これに対し、本論文で提案する方式では、ホスト側の変更は少なく、既存のシステムに適用しやすい事が利点として挙げられる。

本論文では、データベースに於けるアクセスプランが与えられたディスクにおいて、そのアクセスプランを用い、知れたアクセススケジュールによるプリフェッチが行われた場合、どの程度の性能の向上が見込まれるかを評価する為に、データベースのベンチマークであるTPC-Hをアプリケーションとして用い、シミュレーション実験を行った。その結果、大きな実行時間の削減が期待できるという結果が得られた。

以下、2章で従来のディスクアクセスについて簡単に説明し、問題点を明らかにする。その後、アクセスプランを用いたプリフェッチ機構について説明をする。3章では2章で提案したプリフェッチ機構に関する評価実験の結果を述べる。4章では評価実験の結果を示し、考察を行う。最後に、5章で、まとめと今後の課題を述べる。

2 アクセスプランを用いたディスクプリフェッチ

2.1 従来のディスクアクセスとその問題点

アプリケーションの例として、データベースのベンチマークの一つであるTPC-Hを考える。まず、TPC-Hの問合せの一つであるQuery8のSQLとORACLESによって作成されるアクセスプランの一部を図1に示す。

図1のアクセスプランに示されるように、Q8で、ホストは、where句に関して、PARTS,LINEITEM,ORDERSのインデックス、またはテーブルにアクセスし、Nested Loops Joinを行っている。図1に示したSQL文に於けるORACLEシステムのディスクアクセスについて詳しく検討していく。

まず、ORACLEシステムは、

P.TYPE = 'ECONOMY ANODIZED STEEL' (1)

の条件を満たすレコード番号を得る為に、PARTSのインデックスにアクセスを行う。その後、

P.PARTKEY = L.PARTKEY (2)

の条件により、式(1)より得られたレコード番号のPARTSテーブル、LINEITEMのインデックスにアクセスし、NESTED LOOPS JOINを行う。次に、

L.ORDERKEY = O.ORDERKEY (3)

と

O.ORDERDATE (4)

の条件を満たすレコードを見つづける為に、ORDERSのインデックス、テーブルに順次アクセスを行い、式(2)で得られたレコードとNESTED LOOPS JOINを行っている。ORDERKEYに関しては、式(2)で

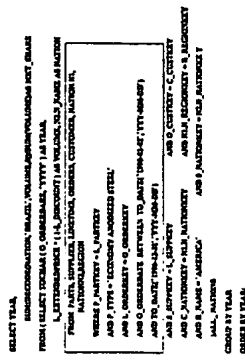


図 1: (左)Query8 of the SQL, (右)Query8 of the SQL

読み込んだLINEITEMのインデックス内にデータが存在していたため、テーブルにはアクセスしていない。
Q8のORACLE8による実際のディスクアクセスの様子を、図2に示す。横軸は実行時間、縦軸はLBA(論理ブロックアドレス)を示している。この図に於いて、プロットした点と点を結ぶ線が、ディスクのシークを表していると考えられることが出来る。

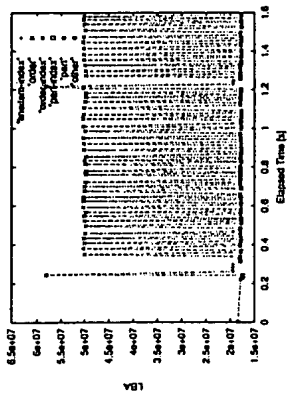


図 2: Q8 of the disk access (initial 1.6ms)

図2からも分かるように、Q8に於いて、ディスクは、ORDERSのインデックスとテーブルに交互にアクセスする。ディスクアクセス時間の支配的要素の一つであるシーク時間の点から考えると、非常に効率的なアクセスが行われていると考えられる。また、ホストは、過去にアクセスしたデータに基づき、ディスクアクセスを行うため、ホストがデータを受け取って、次の命令を出すまでの間、ディスクはストールした状態となっている。(図3)

2.2 アクセスプランベースのプリファETCH

次に、アプリケーションレベルの知識として、アクセスプランがディスクに与えられた場合を考える。こ

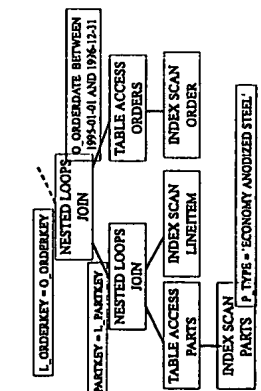


図 3: ディスクのストール

の場合、ディスクは、ORDERSのインデックスを読んだ後、ORDERSのテーブルのどのブロックにアクセスすべきなのか判断できないようになる。従って、ORDERSのインデックスを連続的にアクセスし、そのインデックスによりアクセスすべきテーブルのブロックをスケジューリングし、物理的に近い所からまためてアクセスする事により、シーク時間を短縮する事が出来る。また、このディスクアクセスは、ホストから命令を待つ事なく行われるため、ホストとディスクは、並行して処理作業を行う事ができる。(図4)

図 3: ディスクのストール

さらに、データがディスクアレイのような複数台のディスクに分散して保存されている場合、個々のディスクに並行してアクセスできる為、さらなる実行時間の短縮が期待できる。

3 実ディスクアレイを用いた模擬実験

ディスクにおいてアクセスプラン情報を用いたスケジューリングによるプリファETCHを行った場合、どの程度の性能向上が見込まれるのかを評価するために、ディスクコントローラによるプリファETCHを模擬的に実現するシミュレーションを構築し、評価実験を行う。本来、ディスクコントローラを高機能化し、プリファETCH機構はディスク上に実装されるべきものであるが、その構築

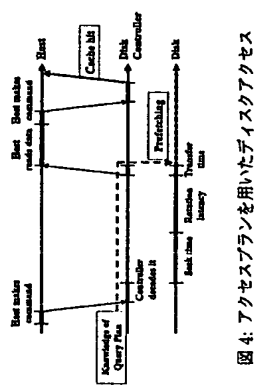


図 4: アクセスプランを用いたディスクアクセス

は困難であることから、ホスト上にプリファETCH機構を模擬的に実行するプリファETCHプロセスを作成し、実行させる。

3.1 実験環境

実験環境を図5に示す。SUN Ultra Enterprise450(4x Ultra SPARC-II 296MHz)をホストとし、ディスクアレイ(Hitachi DF400)を接続する。ディスクアレイにはアクセス用のポートが2つあり、それぞれがホストにWide SCSI インタフェースでつながっている。DBMSにおいて問合せを実行した場合、ディスクアレイには1方のポートからアクセスが行われる。ここで、DBMSがアクセスするポートをDBMSポートと名付け、プリファETCHプロセスがアクセスするポートをprefetchポートと名付ける。ORACLE用のTPC-HのデータサイズはScale Factor=2、すなわち約2G(Byte)で、ディスクアレイのローデバース上にRAID5(4D+1P)で保存されている。ホストは、この2つのポートに対して、個別にコマンドを実行する事ができる。また、ディスクのキャッシュは2つのポートに於いて共有されている。実験に使用したTPC-Hの各テーブルの構成を表1に示す。

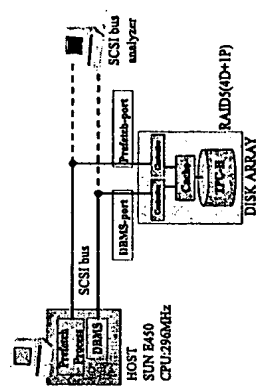


図 5: シミュレーション環境

また、ディスクアレイのそれぞれのポートへのアクセス状況は、SCSIバスアナライザを通し、PCによ

	レコード数	block 数	size(Byte)
PARTS	400K	29370	59M
LINEITEM	12,000K	755350	1,546M
ORDERS	3,000K	193920	376M
PARTSUPP	1600K	125910	258M
SUPPLIER	20K	1590	3M
CUSTOMER	300K	26160	52M
NATION	25	2	4k
REGION	5	1	2k

表 1: 実験用TPC-Hデータセット

	Q8
PARTS.INDEX	56
PARTS	2711
LINEITEM.INDEX	6562
ORDERS.INDEX	41003
ORDERS	68164
OTHER	4913
TOTAL	123409

表 2: 各テーブルへのアクセス数 (TPCH Q8)

り採取することが出来る。

3.2 ディスクアクセスストレー

TPC-Hにおいて、ORACLE システムがQ8を実行した場合のアクセスストレーを、SCSIバスアナライザを用いて採取した。Q8の実行時間は、1180秒であった。また、Q8に於いて行われた各テーブルへのアクセス数を表2に示す。ORACLE システムの最小アクセスブロック単位は2k(Byte)であり、ほとんどのアクセスが最小ブロック単位で行われていた。

3.3 プリファETCHシステム

この節では、ディスクコントローラによるプリファETCHを模擬的に実現するプリファETCHシステムについて、説明を行う。

ORACLE8はDBMSポートを通し、ディスクアクセスを行う。この時、修正されたデバイスドライバによりDBMSポートからの全てのディスクアクセス情報が、アクセスが発生すると同時にプリファETCHプロセスに知らされる。プリファETCHプロセスは、このアクセスを監視し、ORACLEが特定のブロックヘアクセスを行うと、トレーズ情報を基に、prefetchポートを通して、将来、ORACLEが参照すると考えられる

データのプリフェッチを実行する。ディスクアレイ上では、キャッシュが2つのポート間で共有されている。ディスクコントローラがプリフェッチを行った場合と同じ状態が概念的に作成される。プリフェッチシステムを図6に示す。

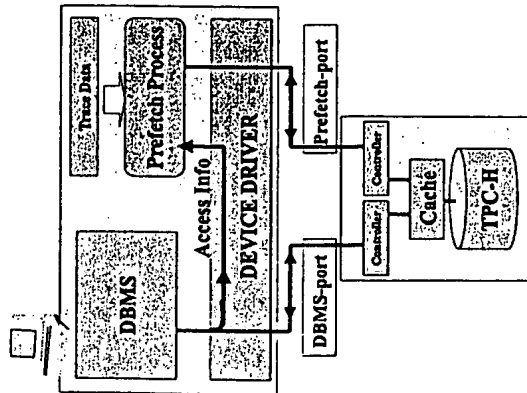


図 6: プリフェッチシステム

3.4 プリフェッチアルゴリズム

この節では、プリフェッチのアルゴリズムについて説明する。アクセスプランにおいて、NESTED LOOPS JOINが行われている部分の取組に注目した時、その取組がインデックスを用いたテーブルアクセスである場合、その取組の部分に於いて、アクセススケジューリングを用いたプリフェッチによるシーケンスの短縮の効果が期待できる。アクセスプランが多重のNESTED LOOPS JOINになっている場合、効率的なスケジューリングを行うという観点からは、より外側のループによるプリフェッチが望ましいが、一度外側のループでデータをORACLEシステムが参照するまでディスクキャッシュから読めないように制限する必要がある。ディスクキャッシュの制限を受ける。よって、一度に行うプリフェッチの量(プリフェッチの深さ)は、統計情報などにより必要なディスクキャッシュ量を予測し、決定される。

Q8を例に取り、説明することにする。図1のアクセス

	実行時間(s)
プリフェッチ無し	1180
LINEITEM.INDEX ベース	423
PART ベース	338
PART.INDEX ベース	233

表 3: Q8 における実行時間の比較 (同時先読み10実行数30)

スプランを考えた場合、Q8は2重のNESTED LOOPS JOINを構成している。最内層は、ORDERS.INDEXを用いたORDERSへのアクセスであり、LINEITEM.INDEX、PARTS、PARTS.INDEXの条件節の基で、アクセスされる。Q8の場合の最も深い先読みはORACLEがLINEITEM.INDEXにアクセスを行ったと同時に、そのLINEITEM.INDEXから得られるORDERS.INDEXとORDERSのプリフェッチを行うことである。Q8の場合のプリフェッチのループの候補は、

1. ORDERS.INDEX, ORDERSのプリフェッチ (LINEITEM.INDEX ベース)
2. LINEITEM.INDEX, ORDERS.INDEX, ORDERSのプリフェッチ (PARTS ベース)
3. PARTS, LINEITEM.INDEX, ORDERS.INDEX, ORDERSのプリフェッチ (PARTS.INDEX ベース)
4. PARTS.INDEX, PARTS, LINEITEM.INDEX, ORDERSのプリフェッチ

となる。ここで、番号が小さい程、深いプリフェッチを示し、番号が大きい程、深いプリフェッチを示している。今回の実験では、Q8に関して、LINEITEM.INDEX ベース(番号1)、PARTS ベース(番号2)、PARTS.INDEX ベース(番号3)のループに関して実験を行った。

4 実験結果

4.1 プリフェッチの深さと実行時間

同時先読み10実行数が30の場合のLINEITEM.INDEX ベース、PARTS ベース、PARTS.INDEX ベースのプリフェッチを行った場合の実行時間を表3に示す。表から明らかな様に、プリフェッチを行わない場合と比較して、プリフェッチを行った場合の実行時間が大きく短縮されていることが判る。また、プリフェッチを行った場合について比較すると、プリフェッチの深さを深くした場合の方がより大幅な性能の向上が示されている。最も深いプリフェッチでは、プリフェッチ無しの場合の5倍以上の性能向上が示されている。

以下、アクセスストレーンを見ながら、より詳細な解析を行う。

4.2 プリフェッチの深さ

この節では、プリフェッチの深さの影響について考える。同時先読み10実行数が30の場合のLINEITEM.INDEX ベース、PARTS ベース、PARTS.INDEX ベースのDBMSポート、prefetchポートからのI/O実行トレースを図7、図8(LINEITEM.INDEX ベース)、図9、図10(PARTS ベース)、図11、図12(PARTS.INDEX ベース)に示す。横軸は時間を示し、縦軸はLBAを示している。

これらの図によって、ORACLEがベースとなるブロックへのアクセスを行ったと同時に、prefetchポートよりプリフェッチが開始されていることが、確認できる。

LINEITEM.INDEX ベースの場合、ORACLEがLINEITEM.INDEXへアクセスを行うと同時に、プリフェッチプロセスはそのLINEITEM.INDEXから得られるORDERS.INDEXとORDERSのプリフェッチを行う。よって、そのブロック群がプリフェッチされた後、プリフェッチプロセスは、ORACLEが次のLINEITEM.INDEXにアクセスを行うまでストールする。図8に於いて、緑色の部分がプリフェッチ機能のストールを示している。次にPARTS ベースの場合について考える。PARTS ベースのプリフェッチを行った場合、今度は、PARTS毎にプリフェッチ機能のストールが起こる(図10)が、LINEITEM.INDEX ベースの場合(図8)と比べると、はるかに少ないことが判る。PARTS.INDEX ベースの場合には、PARTS.INDEX毎にストールするが、そのストールはPARTS ベースの時よりもさらに少なく、図12の枠に入る時間内には一度も発生しない。この様に先読みが深い程、プリフェッチ機能がストールする場合が少なくなり、性能が向上する事になる。

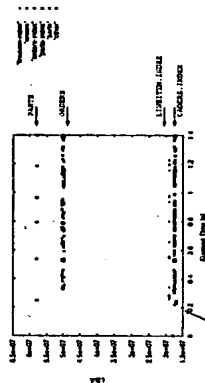


図 7: DBMS ポートのアクセス (LINEITEM.INDEX ベース)

4.3 キャッシュヒット状況

プリフェッチの深さごとのアクセスのヒット状況について検証する。

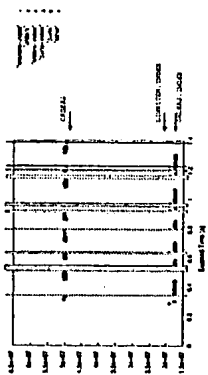


図 8: prefetch ポートのアクセス (LINEITEM.INDEX ベース)

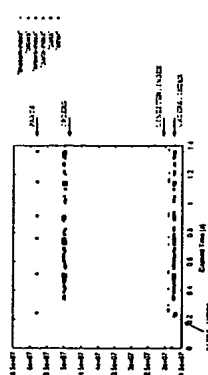


図 9: DBMS ポートのアクセス (PARTS ベース)

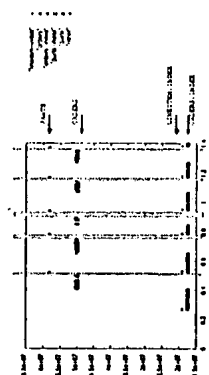


図 10: prefetch ポートのアクセス (PARTS ベース)

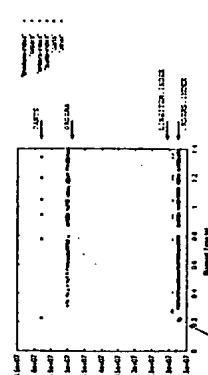


図 11: DBMS ポートのアクセス (PARTS.INDEX ベース)



図 12: prefetch ポートのアクセス (PARTS.INDEX ベース)

前節で示した PARTS ベース、PARTS.INDEX ベースのプリフェッチのヒット状況、ミス状況を、それぞれ、図 13、図 14、図 15、図 16 に示す。

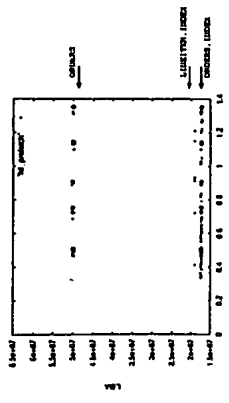


図 13: ヒット状況 (PARTS ベース)

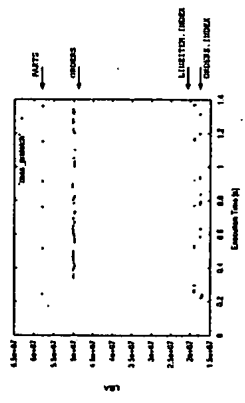


図 14: ミス状況 (PARTS ベース)

図 13、図 15 において、プリフェッチがヒットすることにより I/O 時間が短縮され処理性能が向上している事がプロット点の集中により確認できる。図 16 では、最初の約 0.6 秒間は、DBMS ポートからは、PARTS テーブルと LINEITEM.INDEX のプリフェッチが行われているが、プリフェッチの効果が現れていないが、その後、DBMS ポートのプリフェッチが ORDERS、

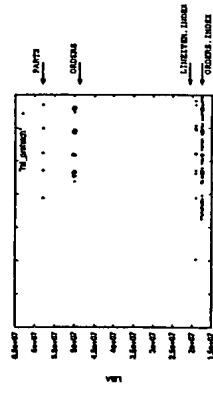


図 15: ヒット状況 (PARTS.INDEX ベース)

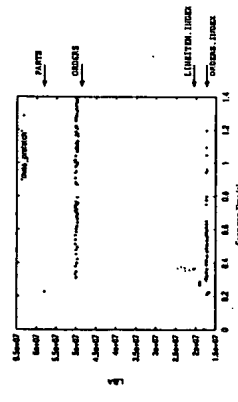


図 16: ミス状況 (PARTS.INDEX ベース)

INDEX と ORDERS に移る為、プリフェッチのヒット率が上昇し、処理時間が短縮されている。表 4 に、それぞれの先読みの深さに関するプリフェッチヒット率を示す。表から明らかな様に、プリフェッチの深さを深くする事により、ヒット率が向上していることが判る。

4.4 同時先読み IO 実行数の影響

一度にディスクがスケジューリングを行う事の出来る同時先読み IO 実行数が、処理時間にどの程度、影響を及ぼすのかを調べる為、同時先読み IO 実行数に対する処理時間をプロットした。その図を図 17 に示す。横軸が同時先読み IO 実行数、縦軸が実行時間で

	ヒット率 (%)
先読み無し	0
LINEITEM.INDEX ベース	77.4
PARTS ベース	82.8
PARTS.INDEX ベース	85.8

表 4: プリフェッチヒット率の比較 (同時先読み IO 実行数 30)

ある。

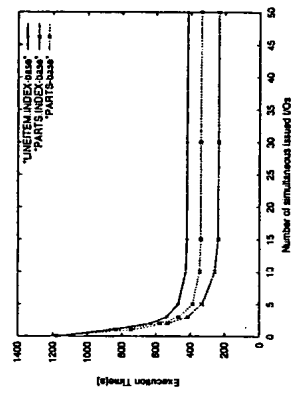


図 17: 同時先読み IO 実行数に対する実行時間

Q8 に於いて、同時先読み IO 実行数が 1 から 10 までの少ない範囲では、実行数が 1 つ増えるごとに処理時間の著しい削減がなされている。これは、ディスクアレイに於いて、ディスク 5 台にデータを分散して保持している為、それぞれのディスクに対し、同時に read コマンドが発行されることによって実現される並列処理による性能向上とスケジューリングの効果によるものと考えられる。しかし、同時先読み IO 実行数が 15 を超えた辺りからは、PARTS.INDEX ベースでは若干の性能向上が見られるが、その他の場合に於いては、処理性能は飽和し、同時先読み IO 実行数の増加に伴う処理時間の向上は見られない。

5 まとめと今後の課題

アプリケーションレベルの知識として、アクセスプランが与えられた場合、そのアクセスプランを用いたプリフェッチ処理を持つディスクに於いてどの程度の性能向上が得られるのかを示す為、類似的なプリフェッチシステムを構築し、評価実験を行った。その結果、アクセスプランによるプリフェッチを行うことにより、ヒット率が向上し、性能が大幅に改善される事が示された。今回は、ORACLE8 に因って実験を行ったが、その他の DBMS においても同じような結果が得られるのか、また、その他の問合せに於いても同様な効果が得られるのかなどの検討が必要と考えられる。今後は、これらの事を踏まえ、さらなる実験を行うていきたい。

謝辞

本研究に御協力いただきました、日立製作所の大枝氏、松浦氏に感謝致します。

参考文献

- [1] D.Patterson et al. "Intelligent RAM(IRAM): the Industrial Setting, Applications, and Architectures". In *Proceedings of the International Conference on Computer Design*, 1997
- [2] Erik Riedel, Garth Gibson, and Christos Faloutsos. "Active Storage For Large-Scale Data Mining and Multimedia". *Proceedings of the 24th VLDB Conference*, New York, USA, 1998
- [3] Anurag Acharya, Mustafa Uysal, and Joel Saltz. "Active Disk: Programming Model, Algorithms and Evaluation". In *Proceedings of ASPLOS VIII*, page 81-91, Oct 1998
- [4] Kimberly Keeton, David A. Patterson, and Joseph M.Hellerstein. "A Case for Intelligent Disks(IDSKs)". *SIGMOD Record*, Volume 27, Number 3, August 1998
- [5] D.A.Patterson, G.Gibson, and R.H.Katz. "A Case for Redundant Arrays of Inexpensive Disks(RAID)". In *Proceedings of ACM SIGMOD*, pp.109-116, Jun, 1988
- [6] A.Acharya, M.Uysal, and J.Saltz. "Active disks". Technical Report, TPC98-06, University of California, Santa Barbara, March 1998
- [7] Mustafa Uysal, Anurag Acharya, and Joel Saltz. "An Evaluation of Architectural Alternatives for Rapidly Growing Datasets: Active Disks, Clusters, SMPs". Technical Report TRCS98-27, University of California, Santa Barbara, Oct 1998
- [8] Cheetah Specification. <http://www.seagate.com/>
- [9] Anurag Acharya, Mustafa Uysal, and Joel Saltz. "Structure and Performance of Decision Support Algorithms on Active Disks". Technical Report, TRCS98-28, University of California, Santa Barbara, Oct 1998
- [10] Aaron Brown, David Oppenheimer, Kimberly Keeton, Randi Thomas, John Kubiakowicz, and David A. Patterson. "ISOTRE: Intrusive Storage for Data-Intensive Network Services". *Proceedings of the 7th Workshop on Hot Topics in Operating Systems(HotOS-VII)*, Rio Rico, Arizona, March 1999.
- [11] <http://www.tpc.org/>
- [12] 梶見聡, 斎藤川原. "高性能ディスクにおけるアクセスプランを用いたプリフェッチ機構の一考察". *DEWS99*